Welcome to the second post of our blog series where we are working on creating a Shiny app with the {golem} package for the hit TV show, The Office. If you are just starting this, please take a look at the first post to see an [overview](https://www.stoltzmanconsulting.com/blog/2020/12/30/r-shiny-golem-development-to-production-overview).

If you simply search “golem” on Google, you see words like “unformed”, “imperfect”, or “shapeless”, but I would not describe the golem package this way. A Shiny app with the golem structure is well defined, organized, and durable. This post walks you through starting a Shiny app with golem, while also highlighting the important files included. This post does not cover everything though, so check out <https://engineering-shiny.org/golem.html> for more information.

Now, let’s get started!

In RStudio, run this in your console:

install.packages(“golem”)

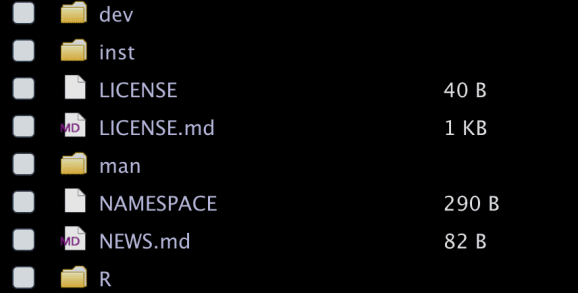
golem::create\_golem("~/Documents/theoffice")

# general case: golem::create\_golem(“PATH/SUBDIRECTORY")

Note: the file location should be adjusted as needed. Once the package is installed, you can also create a golem project by going to ‘File’ → ‘New Directory’ → ‘Package for Shiny App using golem’ in RStudio.

One important note to make about the code above is that we specifically called the golem package to run a function, golem::create\_golem(). Since many packages have conflicting function names, this notation ensures that we use the exact function we want. We will be using this notation throughout the development process.

Right after running this code we are greeted with several folders and files. The folders that we will focus on are R and dev.  
The dev folder contains scripts that will be used frequently:



***01\_start.R***is already open when you start the project and should be completed first. This file allows us to set up our description file, create common files, such as a readme, add a folder for tests and much more.

***02\_dev.R*** enable us to easily add package dependencies, generate modules and add helper functions. This file will be used throughout our development process.

***03\_deploy.R*** is only used when we are ready to deploy our project to a server.

***run\_dev.R*** documents/loads our package and runs the Shiny application locally.  
The R folder comes with the following pre-existing files: ***app\_config.R***, ***app\_server.R***, ***app\_ui.R*** and ***run\_app.R*** . ***app\_server.R*** and ***app\_ui.R*** files function roughly the same as a basic Shiny app’s ***ui.R*** and ***server.R*** .

***run\_app.R*** launches our Shiny app and ***app\_config.R*** is used to access files in the application. Additionally, all of the functions and modules we create for our Shiny app will be stored in the R folder. For a more detailed explanation of these pre-populated files, refer to chapter 4 in <https://engineering-shiny.org/golem.html> .

We will follow these steps to get our project connected to GitHub:

1. Create a new repo on GitHub
   1. ***DO NOT*** initialize with any extra file(s)
2. In the project in RStudio, go up to the terminal and create a new terminal
   1. ***Ensure you are in the right directory*** (where the project is located)
3. Follow the 7 steps from the GitHub repo to “create a new repository on the command line”
   1. echo "# theoffice" >> README.md
   2. git init
   3. Etc.

Once these steps are complete, restart RStudio and you should see a Git tab in the top right panel. Now we’re all set to use GitHub! In addition to these commands, we will add the GitHub repository URL to the description file via ***golem::fill\_desc() i***n  ***01\_start.R.***

You may have noticed ***usethis::use\_git()*** in the ***01\_start.R*** file and ***usethis::use\_github()*** in the ***02\_dev.R*** file. We do not recommend using these functions because you may end up trying to commit thousands of files- not good. The process described above should, hopefully, work every time!

Stay tuned for our next post where we discuss the tools (and tricks) included in these dev files.

“Whenever I’m about to do something, I think, ‘Would an idiot do that?’ and if they would, I do not do that thing.” — Dwight Schrute

Your setup will contain a “***golem::fill\_desc***” section in the ***01\_start.R*** file which allows you to populate the description and package metadata used in the documentation. Below is our initial project code and an example of how ***golem::fill\_desc()*** is filled out.

**R code:**

install.packages(“golem”)

golem::create\_golem("~/Documents/GitHub/theoffice")

**01\_start.R File:**

golem::fill\_desc(

pkg\_name = "theoffice", # The Name of the package containing the App

pkg\_title = "THE\_OFFICE\_GOLEM", # The Title of the package containing the App

pkg\_description = "Shiny Application with golem to Analyze The Office.", # The Description of the package containing the App

author\_first\_name = "Dwight", # Your First Name

author\_last\_name = "Schrute", # Your Last Name

author\_email = "[dwightkschrute@stoltzmanconstuling.com](mailto:dwightkschrute@stoltzmanconstuling.com)", # Your Email

)

There’s not much to your app, however, you can run it. If you do, you’ll see this:

